

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221588436>

A Virtual Tour Guide for Virtual Worlds

Conference Paper · September 2009

DOI: 10.1007/978-3-642-04380-2_40 · Source: DBLP

CITATIONS

53

READS

1,649

5 authors, including:



Anton Leuski

University of Southern California

85 PUBLICATIONS 2,016 CITATIONS

[SEE PROFILE](#)



Jacquelyn Ford Morie

All These Worlds, LLC

93 PUBLICATIONS 1,140 CITATIONS

[SEE PROFILE](#)



David Traum

University of Southern California

319 PUBLICATIONS 10,453 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



RAVE - Robot Avatar Thermal Enhanced Tool [View project](#)



new dimensions in testimony [View project](#)

A Virtual Tour Guide for Virtual Worlds

Dusan Jan, Antonio Roque, Anton Leuski, Jacki Morie, and David Traum

Institute for Creative Technologies, Marina del Rey, CA

Abstract. In this paper we present an implementation of an embodied conversational agent that serves as a virtual tour guide in Second Life. We show how we combined the abilities of a conversational agent with navigation in the world and present some preliminary evaluation results.

1 Introduction

Embodied Conversational Agents (ECAs) allow both verbal and non-verbal interactions [1]. Some ECAs are set within virtual worlds, but few are designed to be able to move around in the virtual world. They are usually placed in a fixed position where the user can interact with them. In this paper we present some of the challenges and our initial solutions for designing conversational agents that can move around in the virtual world and interact with human-controlled avatars and other computer agents.

We have implemented a conversational and navigational agent in Second Life, a virtual world developed by Linden Lab where users can explore, meet other users, socialize and participate in various activities. Our agent, Staff Duty Officer Moleno (SDO), is a junior officer who watches over two islands in Second Life where visitors can find information about the US Army and participate in activities such as a quiz, a helicopter ride, and a parachute jump. As a real staff duty officer would, he patrols the area to make sure everything is ok. Since this is primarily a tourist destination site, he is also equipped to interact with visitors and give them information about the island as well as giving a guided tour as he goes through his rounds. The avatar of SDO Moleno can be controlled either by our agent, or by a human operator.

The SDO's knowledge domain contains information about the two islands. He also is aware of knowledge sources for information such as facts about the army — if visitors ask about this, he tells them to investigate those sources. If someone asks a question that is out of his domain of expertise he promises to try to find the answer, and relays the question to a remote human monitor. In contrast with similar guide or question-answering agents [2–4] the SDO has to navigate the world, handle multiparty situations and has to play a more active role in interactions with visitors. Porting ECAs to Second Life is not new [5–7], but none of these included an ability to guide people through a virtual space.

In the following section, we give an overview of the agent architecture. In Sect. 3, we describe the agent's algorithms for navigating around the island. In Sect. 4 we describe the user modelling component and how different types of

visitors are dealt with in terms of when and how to contact them. In Sect. 5, we describe how conversations are managed. In Sect. 6, we show some examples of conversations visitors have had with SDO Moleno. Finally, in Sect. 7, we present our preliminary evaluation and further evaluation plans.

2 Agent Architecture

The agent is deployed as three separate executables: the main agent application, a text classifier, and a remote control application for human supervisor.

The main agent application is implemented in C# using LibOpenMetaverse (formerly libsecondlife), a .NET based client library used for accessing 3d virtual worlds. This library implements the Second Life client-server protocol and allows for a computer agent to connect to the Second Life virtual world in the same way a human user would connect to it using the Second Life client. The agent application also runs a web server that the remote control applications connect to using TCP protocol. It is used to inform the supervisor about the status of the SDO (whether controlled by computer agent, human operator or offline) and the list of questions that the agent needs help with. In order for the application to determine whether human operator is currently controlling the SDO avatar it connects to Second Life using a separate account while the agent is not running to query the online status of the SDO avatar. This way it can give an accurate status to the supervisor and it knows not to try to automatically connect with the agent when a human is controlling the avatar.

The agent interacts with visitors using text only over chat and instant messages. The understanding of visitor input works purely at the textual level, using cross-language information retrieval techniques to learn the best output for any input from a training set of linked questions and answers. This classification is performed by NPCEditor [8], to which the agent submits the visitor text and replies using the NPCEditor response, postprocessed using contextual information.

3 Navigation

SDO Moleno restricts his movements to two islands which include various buildings and objects of interest. He can move or teleport using kiosks designed to lead to points of interest.¹ Compared to other navigation systems such as [9] we have decided on a relatively simple implementation. Since the domain is fixed we have represented the navigation map as a graph representing the walkways and teleports that the SDO has access to.

SDO Moleno keeps track of which navigation node he is currently closest to. To find a path to a specific location we use the A* algorithm (using the QuickGraph library) from the current location to the navigation node closest to the target. For the final leg to the target the agent just moves directly to target

¹ Flying is also possible, but SDO Moleno avoids flying to make guiding visitors easier.

location under the assumption that the navigation map is designed in a way that the obstacles are accounted for in the graph. If the target is too far away from the navigation map it is treated as inaccessible.

We use two modes for costs of navigation links based on whether the SDO is traveling alone or guiding visitors. In the latter case there is an additional cost associated with teleporting to account for time spent on instructing visitors how to use the teleports and potential problems visitors might have in using them.

One of the features of the SDO is to be able to give a tour of the islands to visitors. This is a partly scripted tour where the SDO gives a brief description at various points of interest, but visitors may also ask questions along the way. The order of the tour is generated dynamically based on where the visitor is at the start using a simple solution of the traveling salesman problem.

4 Visitor Management

One of the main features of the SDO is to proactively assist visitors. Unlike question-answering agents such as [4, 10] where the agent just responds to questions, the SDO actively seeks out the visitors that might need help.

The SDO keeps a user model for everyone he meets. This model persists between sessions. He classifies visitors in several categories that are used in deciding how to interact with them. Each visitor is either marked as a new user, advanced user, computer agent or human supervisor. The SDO also tracks information about visitors to help with decisions, such as last time the visitor was greeted and the time of last interaction with the visitor as well as realtime parameters such as online status, away from keyboard (AFK) status, whether they are typing a message and their location.

The SDO can be in one of several states which determine his behavior. He can be idle, following a path, approaching a visitor, engaged in conversation, guiding visitors or waiting for a visitor. When the SDO is idle he will perform routine rounds, checking if everything on the islands is in order. When he detects a visitor that he is not aware of he will approach their location to investigate.² For identified avatars the SDO will evaluate how important it is to approach them. In general he will only check on new users not marked as AFK to see if they need help, but not more than once every 5 minutes.

When approaching a visitor the SDO will first move to an appropriate distance from the visitor before engaging. The chat has a limit on how far one can see it. In addition the SDO tries to follow proxemics norms as they translate from the real world into virtual worlds [11, 5]. Once he reaches the desired distance. SDO Moleno uses aspects of his information state, including the user model to decide how to address a visitor. If this is the first time he is interacting with a visitor he will introduce himself, give a calling card that enables them to send instant messages to him and offer assistance. For returning visitors he will greet them if he hasn't seen them for some time (more than 3 hours) or just ask if

² Beyond a certain distance, one can only see an avatar's location, not their identity.

everything is ok. A similar greeting behavior is performed when visitors start an interaction as opposed to the SDO seeking them out.

When guiding visitors around the island he makes sure everyone in the group stays with him. If someone falls too far behind he first waits a bit and if they don't come, sends them an instant message, prompting to offer them a teleport if they got lost. During a guided tour, if anyone in the group starts to type the SDO will stop and wait to see what they have to say and enter conversation mode. During conversation if no one takes a turn for a while he will try to resume the guided tour if one is pending.

5 Conversation Management

Visitors can interact with the SDO either in chat or over instant messages (IM). Chat and IM differ in two regards: chat is public but local – everyone who is nearby can see and participate in the chat, but it can only be seen by those who are nearby. IM has no distance limitation, but is private. The SDO can only be in one chat conversation (potentially multiparty) while he can participate in any number of private, dyadic IM conversations going on at the same time. While we have tried to make the SDO human-like in behavior when using chat (e.g., using a typing animation and simulating the time needed to type the messages) we have opted for instantaneous responses in the case of instant messages.

Because of the differences in the media, the SDO responds differently to some content. In some cases visitors will ask the SDO for directions to some location, such as the gift shop. In this case, a good response is not just to answer with directions, but to actually guide the visitor there. This will only work, however, if the visitor is present in the local environment. Moreover, the SDO can only guide to one location at a time, which requires a serialization of interaction that is not required for informational question answering. Because these constraints line up well with the constraints for chat, these kinds of navigational responses are limited to chat conversations. IM conversations will have different responses to direction or action-seeking kinds of initiatives. This difference in chat and IM conversational behavior has been implemented by using several classifiers in NPCEditor[4]: a general one handles generic answers that apply in both situations and each one for chat and IM that inherit from the general one and extend it with answers that only apply in that situation.

When designing question answering agents one usually provides a number of off topic responses to cover the questions that are not understood by the agent [10]. For the SDO we took a different approach since the design goal was to allow a remote supervisor to answer questions that the agent does not have in his domain. Whenever the visitor text is not understood by the classifier we first pass the text to a spell checker. For this purpose we are using NetSpell library. NetSpell's suggestions for a misspelled word are generated using phonetic matching and ranked by a typographical score. In addition we are using a custom domain specific list of words that we use to modify the rank of results based on prior probability of those words appearing in our scenario. This helps us avoid

misunderstandings in presence of typos by the visitor. When spellchecking also does not provide an answer by the classifier, the SDO first gives a prompt asking the visitor to rephrase the question. If he fails to understand two questions in a row he adds the questions to his remote assistance list and tells the visitor he is unable to answer the question, but will try to find out and get back to the visitor. A remote control application allows a human supervisor who can be outside the virtual world but in contact by IM or text message, to monitor the questions that the agent needs help with. The supervisor can either decide to drop the request or provide an answer for it. When the answer is given this is transmitted back to the agent. If he is still in conversation with the visitor he gives them the answer directly (indicating that this is in response to a previous question) or otherwise sends them an IM.

In multiparty conversations we take slightly different approach since we have observed that in many cases the out of domain questions come as a result of two visitors talking to each other about other things. We handle this by delaying the prompt for rephrasing and submission to supervisor. Instead of taking action immediately when he does not understand something SDO Moleno waits until no one is typing for a certain amount of time and only then takes the turn.

The answers that the classifier provides can be templates that include variables representing contextual information and other commands that the agent can interpret. These templates include variables for the current time or name of the addressee, which can be filled in at run-time according to the current context. This can also be used for positive/negative affirmations that need a response based on past context or to provide an answer that depends on contextual knowledge of the agent such as answering a question about who from the group they are waiting for. The annotations are also used for anaphora resolution. For example when a visitor asks “Where is the conference center?” and SDO Moleno replies with “The Army Conference Center is in the SE of Interaction Island.” the answer is annotated with a landmark reference that links to a particular location in the world. This way when a visitor asks “Can you show me how to get there?” SDO Moleno knows how to respond appropriately. In addition the answers are also marked with commands used to generate nonverbal behavior such as gestures and sound as well as instruct the agent to perform specific actions such as guide visitors to a particular location, give them a tour, and offer a teleport.

6 Examples of Interaction

Figure 1 shows an example interaction with a visitor. In this example the interaction was started by the visitor greeting. The SDO identified the visitor as a returning user and returned a greeting appropriate for the current time of day, followed by an offer for assistance. The visitor asks some questions about the events and asks where he could find the helicopter. When the visitor asks the SDO to guide him to the location, SDO Moleno changes into guiding mode, instructs the visitor to follow him and instructs the visitor how to teleport there.



Fig. 1. Example interaction with a returning visitor.

Contrast this dialogue with the following example interaction with a new visitor:

SDO: Good afternoon, I'm Lt Moleno, the Army Island Staff Duty Officer.
 SDO: Here is a calling card if you ever need to call me.
 SDO: How can I help you?
 Visitor: i am just looking around
 SDO: Have a nice stay, please feel free to explore and try out our interactive events.

The following sample interaction shows an example of how the SDO handles an out of domain question that needs supervisor input:

Visitor: when will the island open for public?
 SDO: Could you rephrase that?
 Visitor: can you tell me when others will be able to come here?
 SDO: I dont know, but I will find out and get back to you.
 ...
 SDO: I found the answer to what you were asking me before.
 SDO: We will be opening the islands for an education fair on April 29.
 Visitor: thank you

7 Preliminary Evaluation & Plans

During the design of the agent we have performed an informal evaluation of the conversational capabilities of SDO Moleno in order to determine where we need to make more improvements. We had several users interact with the agent with a domain of about 300 questions and 70 answers. We have used two coders to annotate the questions into categories indicating what kind of response the agent should have given (in domain, no response needed, off topic, relay to supervisor).

The sample included 155 questions with agreement of 68% and a kappa score of 0.37 for the two coders. The average agreement between actual agent responses and the coders was 47% with kappa score 0.22. One main distinction we noticed was that the SDO relayed a lot more questions to the supervisor than expected. We have used this data to enhance the domain coverage and change the threshold parameters of the classifier. Once the islands open we plan a more extensive evaluation of the SDO performance and visitor satisfaction.

Acknowledgments. This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and the content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

1. Cassell, J., Sullivan, J., Prevost, S., Churchill, E., eds.: Embodied Conversational Agents. MIT Press, Cambridge, MA (2000)
2. Gustafson, J., Lindberg, N., Lundeberg, M.: The august spoken dialogue system. In: Proceedings of Eurospeech 99. (1999)
3. Kopp, S., Gesellensetter, L., Krämer, N.C., Wachsmuth, I.: A conversational agent as museum guide - design and evaluation of a real-world application. In: IVA. Volume 3661 of LNCS., Springer (2005) 329–343
4. Leuski, A., Patel, R., Traum, D., Kennedy, B.: Building effective question answering characters. In: Proceedings of SIGdial Workshop. (2006) 18–27
5. Friedman, D., Steed, A., Slater, M.: Spatial social behavior in second life. In: IVA. Volume 4722 of LNCS., Springer (2007) 252–263
6. Endraß, B., Prendinger, H., André, E., Ishizuka, M.: Creating and scripting second life bots using mpml3d. In: IVA Volume 5208 of LNCS, Springer (2008) 492–493
7. Weitnauer, E., Thomas, N.M., Rabe, F., Kopp, S.: Intelligent agents living in social virtual environments - bringing max into second life. In: IVA Volume 5208 of LNCS. (2008) 552–553
8. Leuski, A., Traum, D.: A statistical approach for text processing in virtual humans. In: 26th Army Science Conference, Orlando, Florida. (2008)
9. Gayle, R., Manocha, D.: Navigating virtual agents in online virtual worlds. In: Proceedings of the Web3D 2008, ACM New York, NY, USA (2008) 53–56
10. Artstein, R., Cannon, J., Gandhe, S., Gerten, J., Henderer, J., Leuski, A., Traum, D.: Coherence of Off-Topic Responses for a Virtual Character. In: 26th Army Science Conference, Orlando, Florida. (2008)
11. Jan, D., Traum, D.: Dynamic movement and positioning of embodied agents in multiparty conversations. In: proceedings of AAMAS 2007. (2007) 59–66